Assignment 5: Data Visualization

Aileen Lavelle

Spring 2023

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on Data Visualization

Directions

1. Rename this file `<FirstLast>_A05_DataVisualization.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure your code is tidy; use line breaks to ensure your code fits in the knitted output.
5. Be sure to **answer the questions** in this assignment document.
6. When you have completed the assignment, **Knit** the text and code into a single PDF file.

Set up your session

1. Set up your session. Load the tidyverse, lubridate, here & cowplot packages, and verify your home directory. Upload the NTL-LTER processed data files for nutrients and chemistry/physics for Peter and Paul Lakes (use the tidy `NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv` version) and the processed data file for the Niwot Ridge litter dataset (use the `NEON_NIWO_Litter_mass_trap_Processed.csv` version).

2. Make sure R is reading dates as date format; if not change the format to date.

```
#1
library(tidyverse); library(lubridate); library(here); library(cowplot);
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
##
## Attaching package: 'lubridate'
```

```
##
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
##
##
## here() starts at /Users/aileen/Desktop/Duke/Environmental_Data_Analytics/EDA_Spring_2023_corrected
##
##
## Attaching package: 'cowplot'
##
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```r
here()
```

```
## [1] "/Users/aileen/Desktop/Duke/Environmental_Data_Analytics/EDA_Spring_2023_corrected"
```

```r
PeterPaul.chem.nutrients <-
  read.csv(here("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
           stringsAsFactors = T)
PeterPaul.chem.physics <-
  read.csv(here("Data/Processed_KEY/NTL-LTER_Lake_ChemistryPhysics_PeterPaul_Processed.csv"),
           stringsAsFactors = T)
NIWOTRidge <-
  read.csv(here("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
           stringsAsFactors = T)

#2
#setting date columns to be read as a date using lubridate
NIWOTRidge$collectDate <- ymd(NIWOTRidge$collectDate)
PeterPaul.chem.nutrients$sampledate <- ymd(PeterPaul.chem.nutrients$sampledate)
PeterPaul.chem.physics$sampledate <- ymd(PeterPaul.chem.physics$sampledate)
```

Define your theme

3. Build a theme and set it as your default theme. Customize the look of at least two of the following:

- Plot background
- Plot title
- Axis labels
- Axis ticks/gridlines
- Legend

```r
#3
Aileentheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        axis.ticks = element_line(color = "black"),
        plot.background = element_rect(color= "white"))
```

Create graphs

For numbers 4-7, create ggplot graphs and adjust aesthetics to follow best practices for data visualization. Ensure your theme, color palettes, axes, and additional aesthetics are edited accordingly.

4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

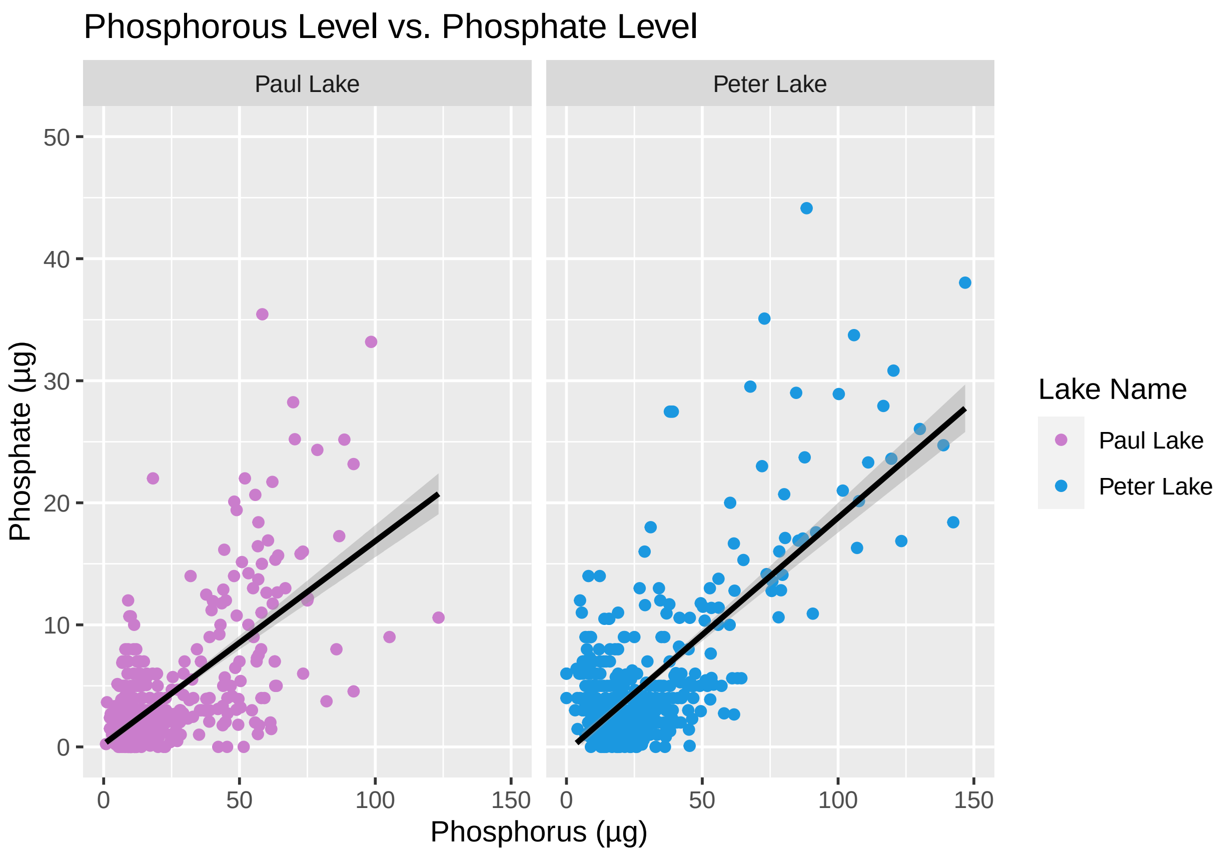
```
#4
PhosphorusbyPhosphate <-
  ggplot(PeterPaul.chem.nutrients, aes(x=tp_ug, y=po4)) +
  geom_point(aes(color= lakename)) +
  geom_smooth(formula =y ~ x, color= "black") +
  xlim(0, 150) +
  ylim(0, 50) +
  facet_wrap(vars(lakename)) +
  labs(
    title = "Phosphorous Level vs. Phosphate Level",
    x= "Phosphorus (µg)",
    y= "Phosphate (µg)",
    color= "Lake Name") +
  scale_color_manual(values = c("#ca7dcc",
                                "#1b98e0"))
PhosphorusbyPhosphate
```

```
## ‘geom_smooth()‘ using method = ’gam’

## Warning: Removed 21948 rows containing non-finite values (‘stat_smooth()‘).

## Warning: Removed 21948 rows containing missing values (‘geom_point()‘).

## Warning: Removed 2 rows containing missing values (‘geom_smooth()‘).
```



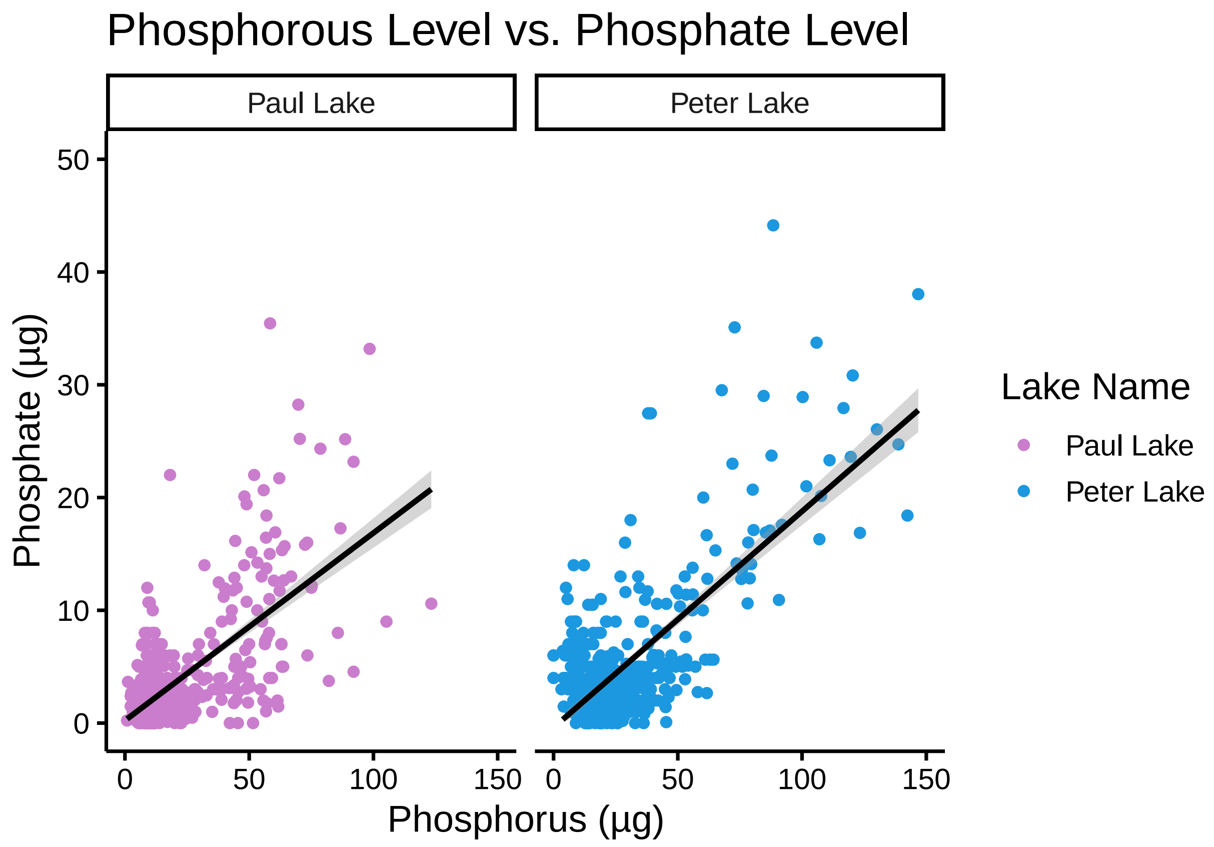
```
#Plotting phosphorus by phosphate with my theme
PhosphorusbyPhosphate + Aileentheme
```

```
## `geom_smooth()` using method = 'gam'

## Warning: Removed 21948 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 21948 rows containing missing values (`geom_point()`).

## Warning: Removed 2 rows containing missing values (`geom_smooth()`).
```

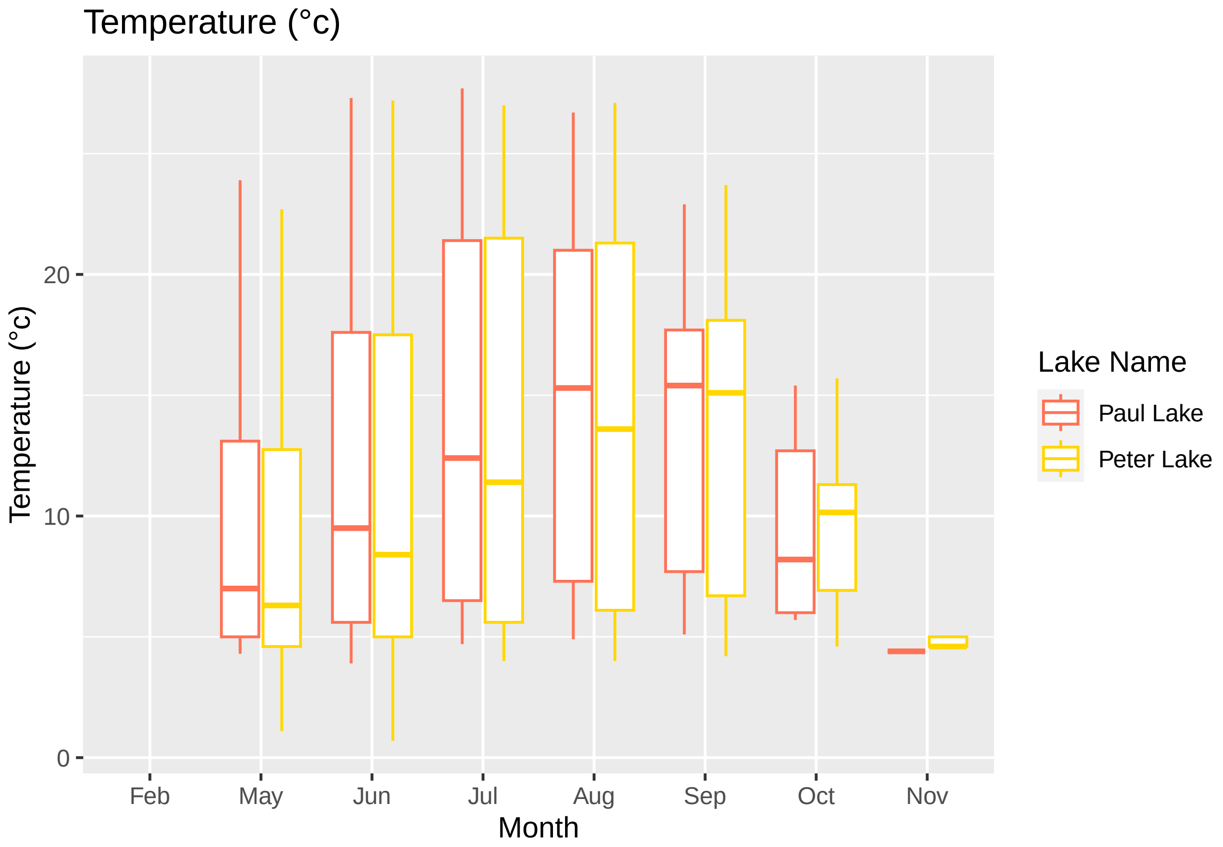


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

Tip: R has a build in variable called `month.abb` that returns a list of months;see https://r-lang.com/month-abb-in-r-with-example

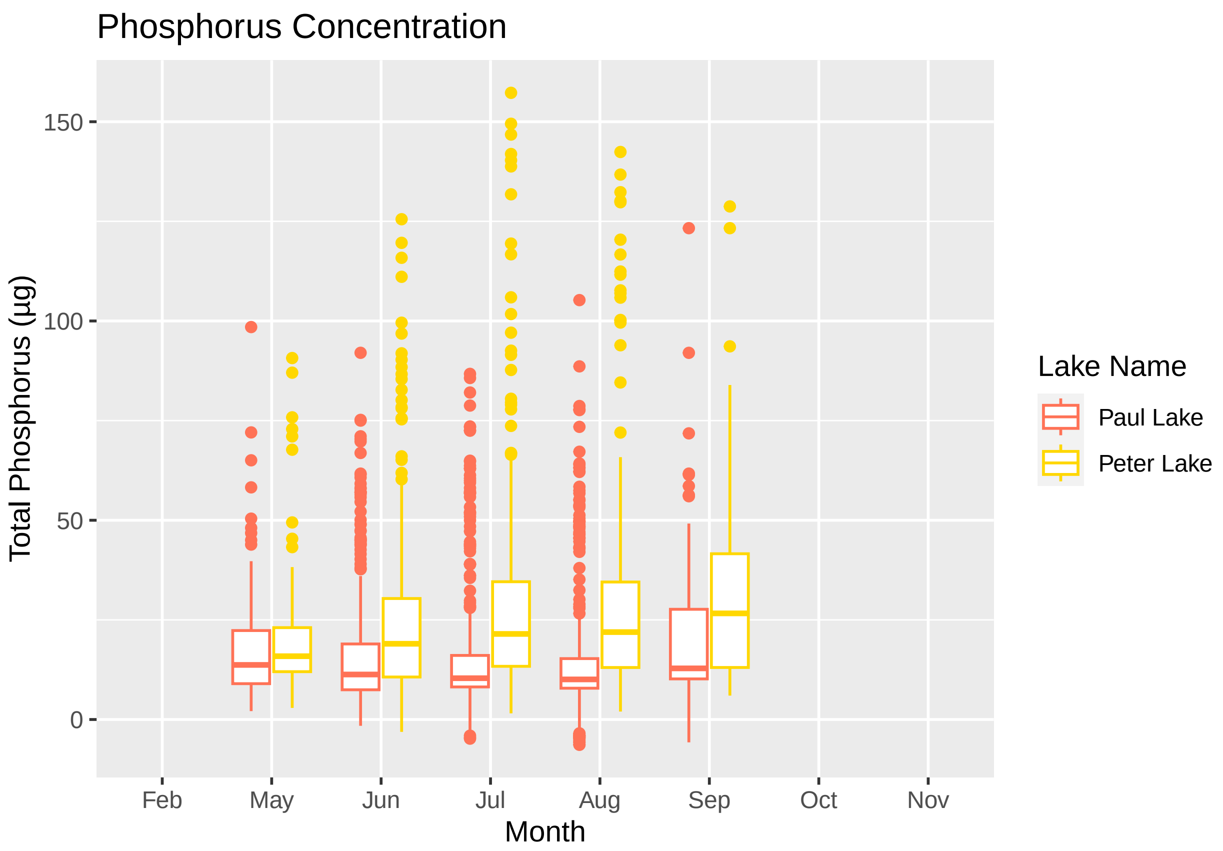
```
#5
#create each plot seperately
Temp <-
  ggplot(PeterPaul.chem.nutrients, aes((x=factor(month, levels=1:12,
  labels=month.abb)), y=temperature_C)) +
  geom_boxplot(aes(color= lakename)) +
  labs(
    title = "Temperature (°c)",
    x= "Month",
    y= "Temperature (°c)",
    color= "Lake Name") +
  scale_color_manual(values = c("#FF7256",
                                "#FFD700"))
Temp
```

```
## Warning: Removed 3566 rows containing non-finite values (‘stat_boxplot()‘).
```



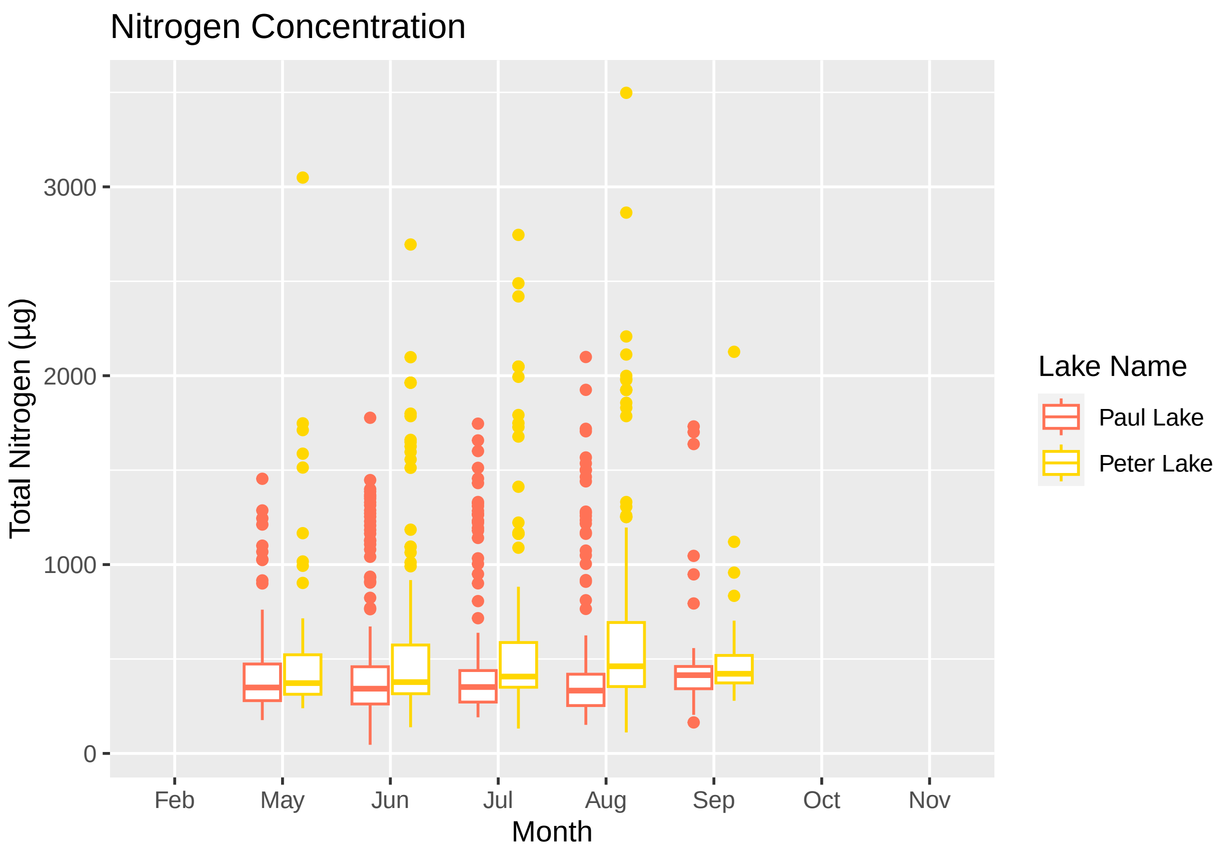
```
Tp <-
  ggplot(PeterPaul.chem.nutrients, aes((x=factor(month, levels=1:12,
  labels=month.abb)), y=tp_ug)) +
  geom_boxplot(aes(color= lakename)) +
  labs(
    title = "Phosphorus Concentration",
    x= "Month",
    y= "Total Phosphorus (µg)",
    color= "Lake Name") +
  scale_color_manual(values = c("#FF7256",
                                "#FFD700"))
Tp
```

```
## Warning: Removed 20729 rows containing non-finite values (`stat_boxplot()`).
```



```
Tn <-
   ggplot(PeterPaul.chem.nutrients, aes((x=factor(month, levels=1:12,
  labels=month.abb)), y=tn_ug)) +
  geom_boxplot(aes(color= lakename)) +
  labs(
    title = "Nitrogen Concentration",
    x= "Month",
    y= "Total Nitrogen (µg)",
    color= "Lake Name") +
  scale_color_manual(values = c("#FF7256",
                                "#FFD700"))
Tn
```

```
## Warning: Removed 21583 rows containing non-finite values (`stat_boxplot()`).
```

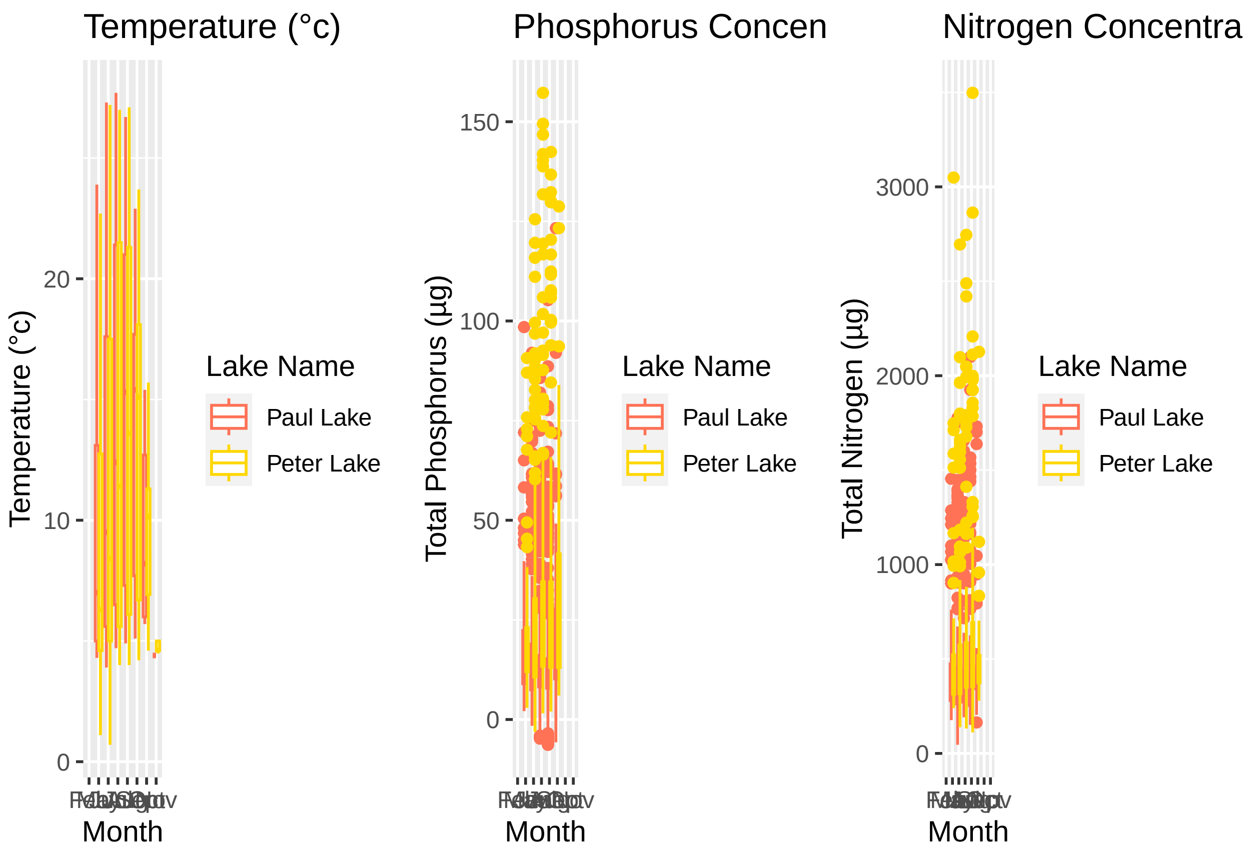


```
#combine with cowplot
plot_grid(Temp, Tp, Tn, nrow=1,labels=c(''))
```

```
## Warning: Removed 3566 rows containing non-finite values (‘stat_boxplot()‘).

## Warning: Removed 20729 rows containing non-finite values (‘stat_boxplot()‘).

## Warning: Removed 21583 rows containing non-finite values (‘stat_boxplot()‘).
```



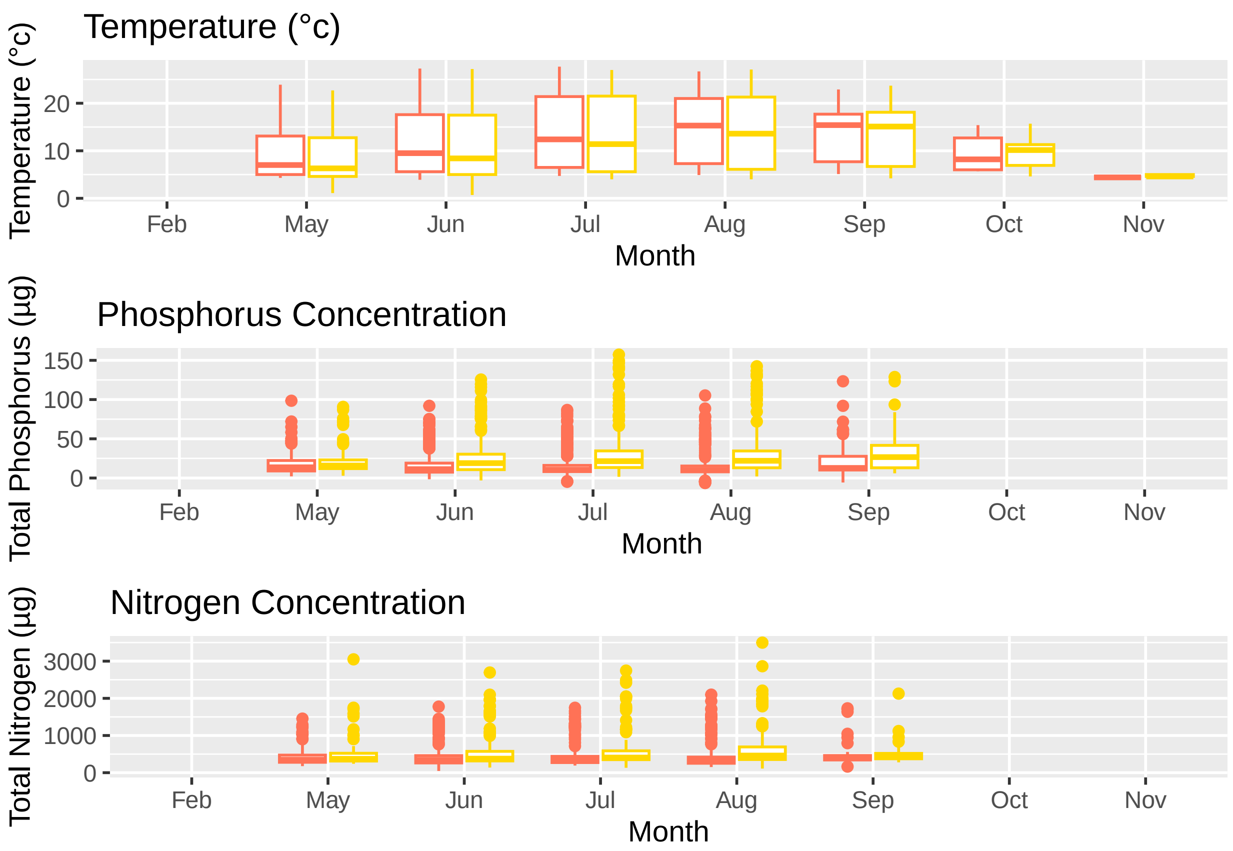
```
#remove duplicate legend
Temp = Temp + theme(legend.position="none")
Tp = Tp + theme(legend.position="none")
Tn_new = Tn + theme(legend.position="none")
# Homogenize scale of shared axes
Temp_tn_tp <-
  plot_grid(
  Temp, Tp, Tn_new,
  align = "h", axis = "b", nrow = 3, rel_widths = c(1, 2, 3)
)
```

```
## Warning: Removed 3566 rows containing non-finite values (‘stat_boxplot()‘).

## Warning: Removed 20729 rows containing non-finite values (‘stat_boxplot()‘).

## Warning: Removed 21583 rows containing non-finite values (‘stat_boxplot()‘).
```

```
Temp_tn_tp
```

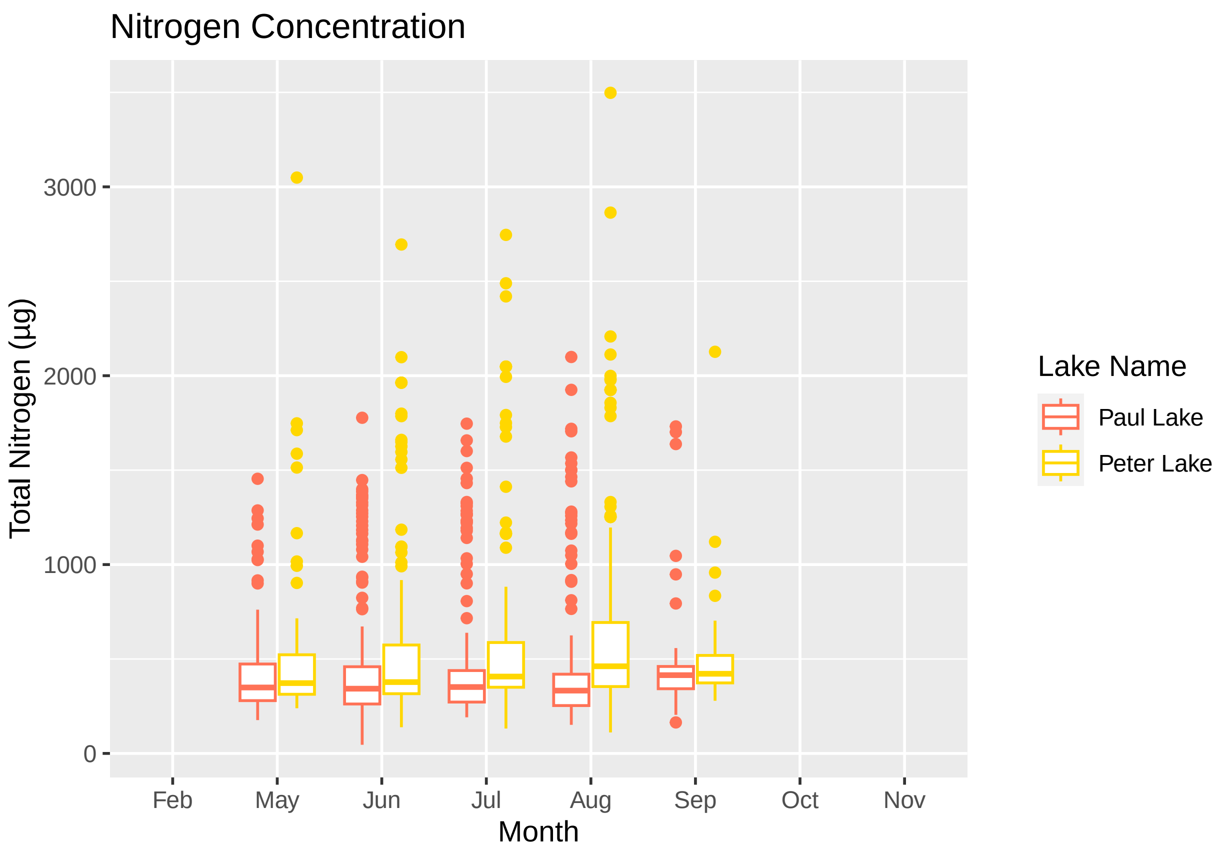


```
# extract the legend from one of the plots
legend <- get_legend(Tn)
```

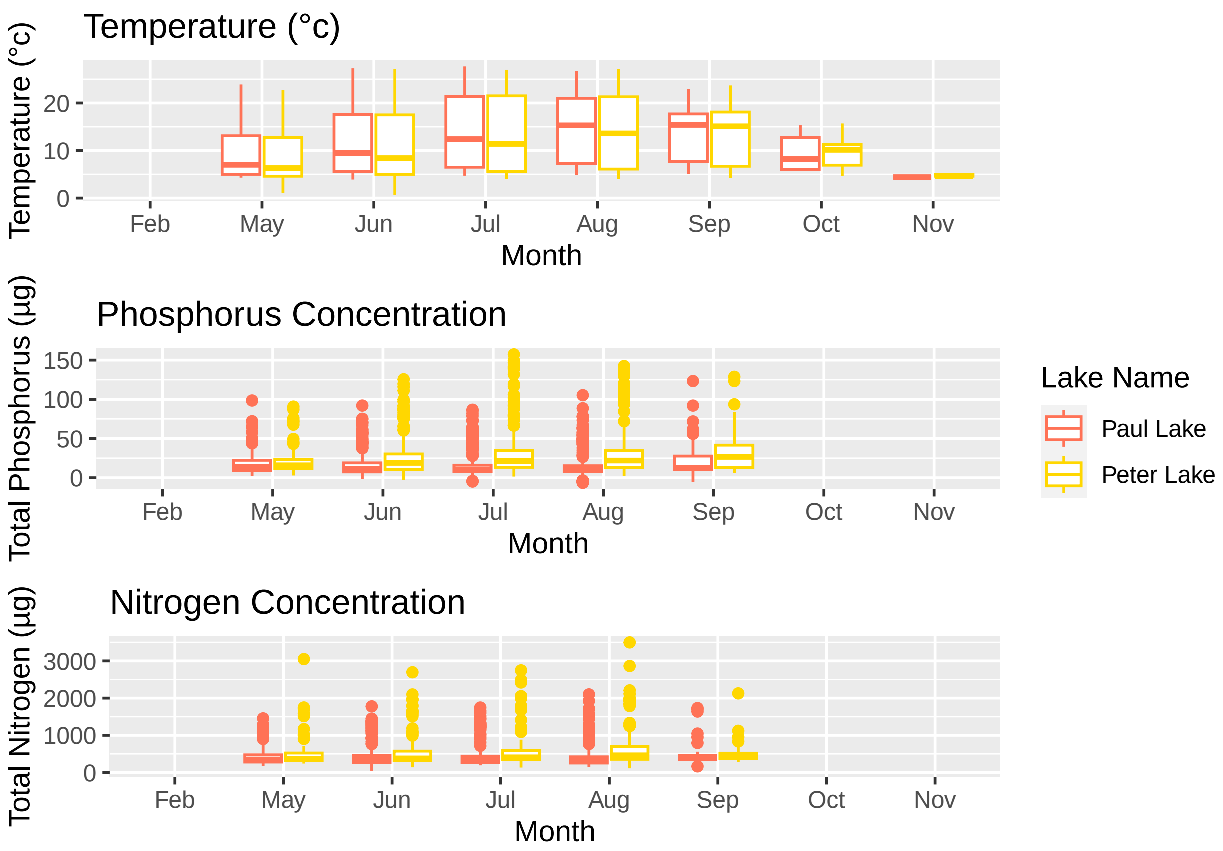
```
## Warning: Removed 21583 rows containing non-finite values (`stat_boxplot()`).
```

```
# create some space to the left of the legend
Tn + theme(legend.box.margin = margin(0, 0, 0, 10))
```

```
## Warning: Removed 21583 rows containing non-finite values (`stat_boxplot()`).
```



```
# add the legend to the row we made earlier. Give it one-third of
# the width of one plot (via rel_widths).
plot_grid(Temp_tn_tp, legend, rel_widths = c(.9, .2))
```



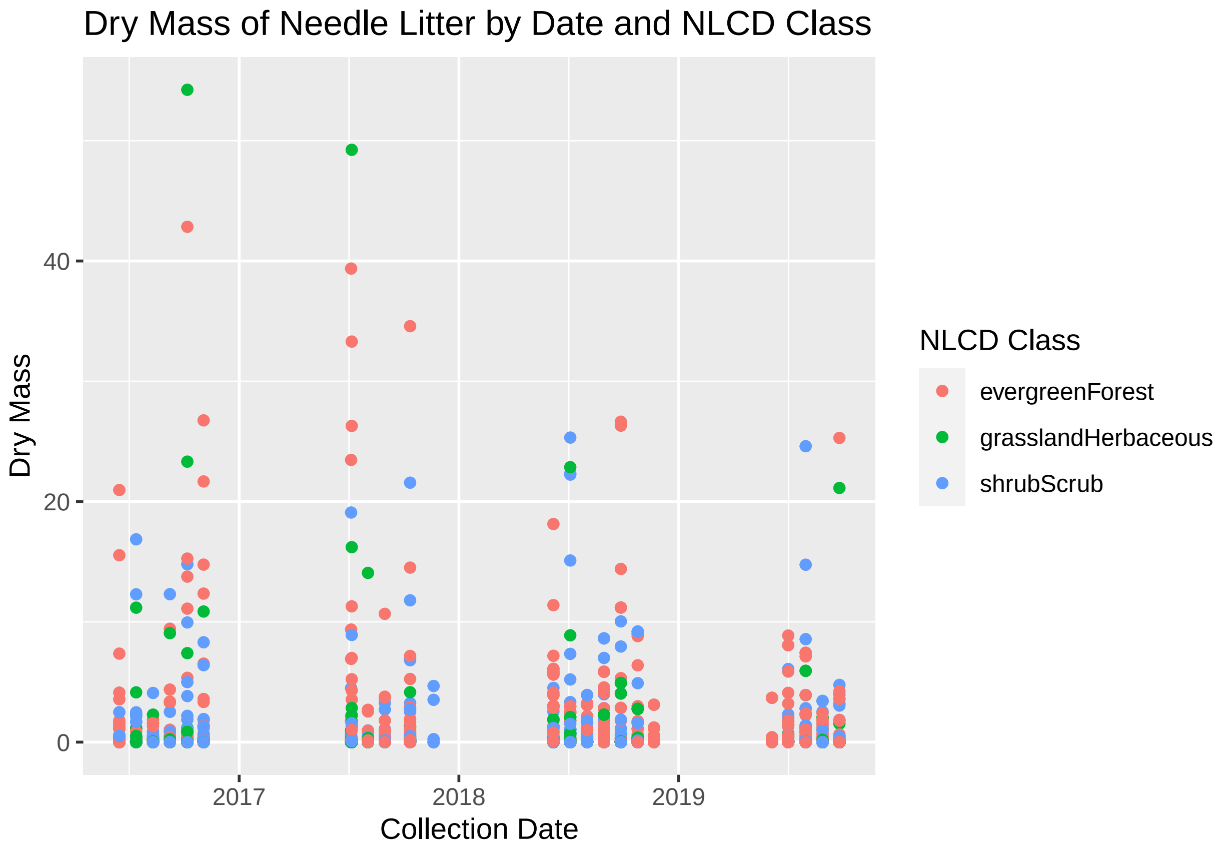
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: The temperature reaches a high in the summer, between June through September. The variance within the temperature boxplots is generally large, except for November, which has a tight spread for both lakes. Phosphorus is slightly higher in Peter Lake than in Paul Lake. There is no data on total phosphorus or Nitrogen levels in February, October, December, or January. The Phosphorus and Nitrogen levels are skewed toward lower concentrations, but large outliers are present throughout all the months in both lakes. The total nitrogen and phosphorus concentrations in Peter Lake are slightly higher than for all months in Paul Lake. Total Nitrogen levels in Peter lake were greatest in August and in the summer months. Phosphorus levels are highest in Peter lake in the summer, June-September. In Paul lake, the phosphorus was most elevated in May and September, and the lack of other data indicates that the summer also has the highest phosphorus levels in Paul lake.

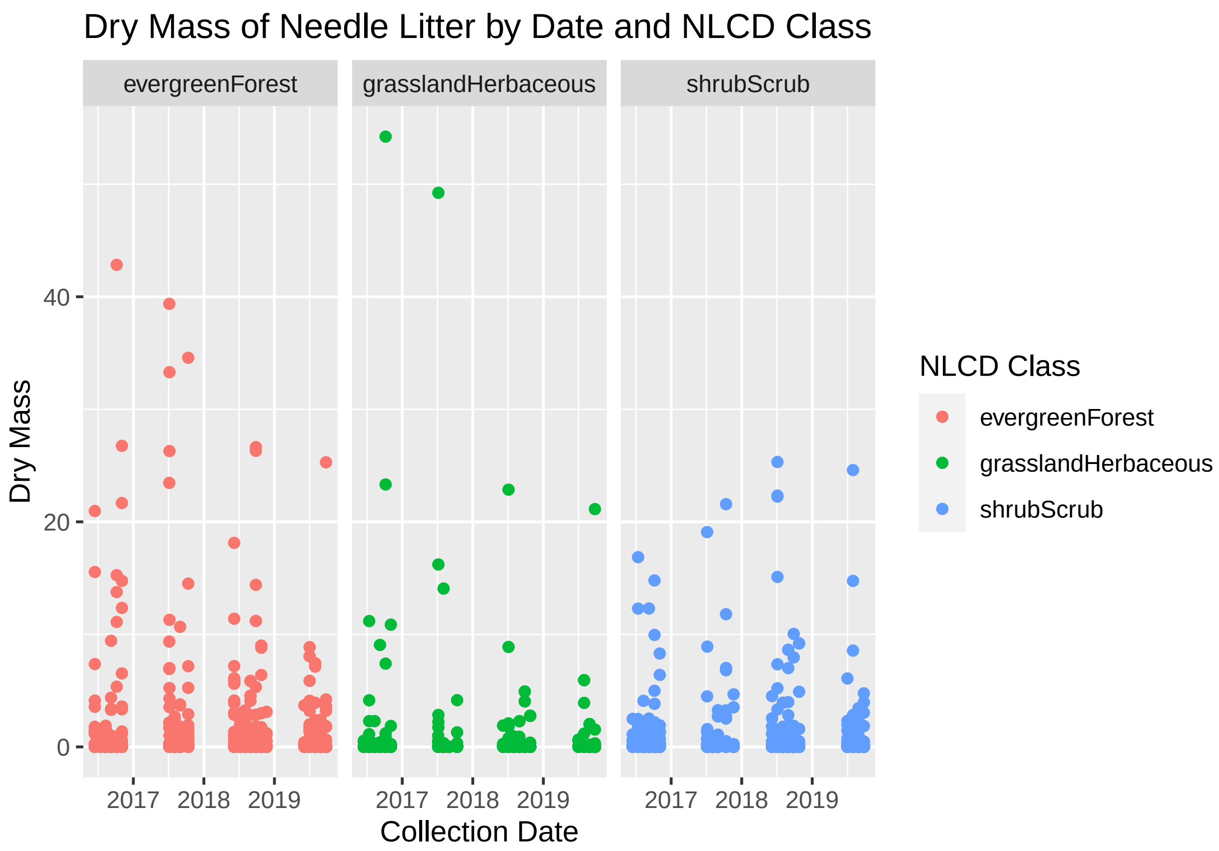
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the “Needles” functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6
#filter for only the Needles functional group
Needles <- NIWOTRidge %>%
   filter(functionalGroup %in% c("Needles"))
#create plot with NLCD class coloring
```

```
Needles <- NIWOTRidge %>%
  ggplot(aes(x= collectDate, y=dryMass, color= nlcdClass)) +
  geom_point() +
  labs(
    title = "Dry Mass of Needle Litter by Date and NLCD Class",
    y= "Dry Mass",
    x= "Collection Date",
    color = "NLCD Class")
Needles
```



```
#7
#separate into facets
Needles + facet_wrap(vars(nlcdClass))
```



Question: Which of these plots (6 vs. 7) do you think is more effective, and why?

> Answer: I think that plot 7 looks better because you can more easily see all of the points for the different NLCD classes. When the NLCD classes are not seperated into three facets there is a lot of overlap of points and it is harder to differentiate and thefore see patterns in the data. I also think that it nicely lines up the dry masses so you can see which NLCD class has more outliers and then compare by year.